UNITED STATES PATENT AND TRADEMARK OFFICE

A

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/896,179	06/29/2001	Lin Yu	9209-7	9539

20792 7590 07/28/2005

MYERS BIGEL SIBLEY & SAJOVEC
PO BOX 37428
RALEIGH, NC 27627

EXAMINER	
CHEN, TE Y	

ART UNIT	PAPER NUMBER
2161	

DATE MAILED: 07/28/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/896,179
Filing Date: June 29, 2001
Appellant(s): YU, LIN

Scott Moore
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 08/18/2004.

(1) Real Party in Interest

A statement identifying the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

A statement identifying the related appeals and interferences which will directly affect or be directly affected by or have a bearing on the decision in the pending appeal is contained in the brief.

(3) Status of Claims

The statement of the status of the claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Invention

The summary of invention contained in the brief is correct.

(6) Issues

The appellant's statement of the issues in the brief is correct.

(7) Grouping of Claims

Appellant's brief includes a statement that Group I (Claims 1, 2, 8, 14, 15, 21, 28, 29, 35 and 36) and Group II (Claims 3-7, 9-13, 16-20, 22-27, 30-34, and 37-42) do not stand or fall together and provides reasons as set forth in 37 CFR 1.192(c)(7) and (c)(8).

(8) Claims Appealed

The copy of the appealed claims contained in the Appendix to the brief is correct.

9) Prior Art of Record

5,204,958	Cheng et al.	Apr. 20, 1993

The following is a listing of the prior art introduced with instant Examiner's Answer

The definition of the phrase "burst" in Microsoft Press Computer Dictionary, 3rd Edition (published in 1977).

(10) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-42, are rejected under 35 U.S.C. 102(e) as being anticipated by anticipated by Cheng at al. (U.S. Patent No. 5,204,956).

As to claims 1-5, 9-11, 14-18, 22-24, 27-32, and 37-39, Cheng at al. (hereinafter referred as Cheng) discloses a system with methods / means / computer program product to perform the following functions:

a) storing the spaced apart bursts of data records in a database [e.g. the step 300, Fig. 4A; col. 2, lines 60-62, 65-67; col. 6, lines 48-59].

b) deferring the build of an index for a corresponding one of the spaced apart bursts until the corresponding one of the series of spaced apart records being stored in the database [e.g. the steps 300-302, Fig 4A; col. 2, lines 45-48; col. 6, lines 60-64].

As to claims 6-7, 12-13, 19-20, 25-26, 33-34 and 40-41, Cheng further discloses that the system can be configured to either store / build the index on either a different or single processor [col. 2, lines 51-57].

As to claims 8, 21, 35-36 and 42, Cheng further discloses that the system is configured to provide a Indexed Sequential Access Method (ISAM) database [e.g., col. 2, lines 65 – col. 3, line 16].

(11) Response to Argument

Applicant's arguments filed on 08/18/2004 have been fully considered but they are not persuasive.

For Group I: Examiner disagrees with Appellant's main arguments that "Cheng does not describe any solution for storing temporarily spaced apart bursts of data records in a database as recited in claims 1, 14 and 28"

In reply to these arguments: Examiner first noted that Appellant refers to "bursty" data of Claims 1, 14 and 28 as describe in the instant Appeal Brief (at Page 4) is misleading because these claims only recite "burst of data" not "bursty data" and the definition of "burst" provided in the Microsoft Computer Dictionary, 3rd Edition (published in 1977) is "transfer of a block of data all at one time without a break".

In addition, Examiner counters by pointing out that Cheng specifically discloses a database management system comprising the claimed features at col. 6, lines 48-64 as following:

"Referring to FIGS. 3 and 4, the database management program 146 works as follows. Whenever a new data record is received, it is stored in secondary memory in the sequential data file 130 (step 300). This is called a "sequential" data file because new records are always written to the end of the file in a temporal order, regardless of their index value. In some systems, a small number of such records may be temporarily stored in a buffer 250 until a full page, or several full pages, of such records is ready to be stored, or until a transaction "commits", requiring that all records related thereto be securely stored.

"After a number of records are stored in secondary memory file 130, a block of these records are read, and corresponding indexed pointer are created and temporarily stored in primary memory in the index file 148 (step 302), herein called the small B-tree (SBT) 148."

Wherein, the claimed feature "storing temporally spaced apart bursts of data records" read by Cheng's description as "all number of such records may be temporarily stored in a buffer 250 until a full page, or several full pages, of such records is ready to be stored, or until a transaction "commits". " Because all number of such data records is temporally stored in the buffer 250, and space apart by page block such as "until a full

page" or "until several full pages" of such records are ready to be stored, or "until a transaction is committed", then requiring that all records related thereto be securely stored in a database system [e.g., the unit 130, Fig. 3].

Moreover, Cheng clearly discloses storing all these spaced apart bursts of records in a database [e.g., the unit 130, Fig. 3].

Furthermore, Cheng specifically discloses deferring building the index for the corresponding one of the spaced apart burst of records after storing these spaced apart burst of records at col. 6, lines 48-64 as recited above.

In addition, Cheng further discloses the following:

"Periodically, a portion of the memory resident small B-tree is merged with a corresponding portion of the large B-tree by selecting a range of index
values and retrieving from secondary memory all indexed pointers in the selected range of index values. The indexed pointers in the first B-tree in the selected range of index values are merged into the retrieved records, the resulting merged set of indexed pointers are stored in secondary memory in indexed order in a contiguous area of secondary memory at the tail of the large B-tree. As a result, the indexed pointers for newly added database records are written to secondary memory in batches, thereby accessing secondary memory very efficiently." (col. 3, lines 4-16)

"The purpose of the SBT 150 is to temporarily store indexed pointers, and to store a sufficiently large number of these pointers to enable efficient storage of these indexed pointers in secondary memory using a rolling merge type of procedure. For the purpose of explanation, we will assume that the SBT 150 typically holds about 20,000 indexed pointers at any one time, representing about 200 seconds of record insertions (assuming that records are inserted on average at 100 records per second). Using a three level tree with 100 items stored in each node, the SBT 150 could store up to one million indexed pointers. Since only about 20,000 will be stored, the second level non-leaf nodes will be very sparsely populated. On the

other hand, if there is a temporary surge of record insertions, or if the disks in secondary storage were unavailable for a period of ten minutes (e.g., for making a tape backup of secondary storage), the three-level SBT can easily absorb a much larger number of indexed pointers so long as sufficient primary memory is available." (col. 7, lines 18-37)

"The process of merging the SBT 148 into the LBT is done in stages, with steps 306 to 316 being repeated many times. For example, every two seconds the
system might merge one percent of the indexed pointers in the SBT 148 into the LBT 132. To do this, the DBMS program 146 selects a range of index values I1 to I2, copies all the indexed pointers in the SBT within that range into a buffer 252 (which is just an array in primary memory) called Buffer A. A pointer ST keeps track of the lowest index value not yet copied into Buffer A (step 306). All the indexed pointers in the SBT 148 are retained at this time so that a search for any of these items can be performed prior to completing the merger of the SBT 148 into the LBT 132..." (col. 7, lines 59 -col. 8, lines 46)

Wherein these paragraphs reveals another way for deferring building indexes after storing bursts of index records in the Small B-Tree (SBT). For example, periodically using a rolling merge type of procedure (col. 7, line 21-22) to sparsely populated (col. 7, lines 29-31) a sufficiently large number of these pointers (or burst of records) of the Small B-Tree (SBT 150, Fig. 3) in batches (col. 3, line 15) or in units of pages (col. 8, line 9) into a Large B-Tree (LBT 132, Fig. 3) of a secondary memory (the unit 106, Fig. 1).

Thus, base on the discussion above, in contrary to Appellant's arguments the features in claims 1,14 and 28 are fully anticipated by Cheng.

For Group II: Appellant's arguments that "Cheng does not appear to provide any description or suggestion of storing temporarily spaced apart bursts of data record that

are received during a corresponding series of spaced apart time intervals. Nor does Cheng appear to describe or suggest storing the spaced apart of data records in the database during the corresponding series of spaced apart time intervals. Finally, Cheng does not appear to describe or suggest beginning to build the index for a corresponding one of the spaced apart bursts after expiration of the corresponding one of the series of spaced apart time intervals, as recited in Claims 9, 22 and 37." Examiner is not persuaded.

Careful consideration of Cheng's invention reveals that the storing of claimed spaced apart bursts of data records read by the time intervals associated to the temporary storing such records in a buffer 250 at col. 6, lines 54-59 of Cheng's disclosure, until the time elapse for "a full page" or "several full pages, of such records are ready to be stored, or until a "transaction commits" in a database system [e.g., the unit 130, Fig. 3].

In addition, Cheng clearly discloses the storing of all these spaced apart burst of records in a database [e.g., the unit 130, Fig. 3] via the step performed by the unit 300, Fig. 4A and associated texts.

Furthermore, Cheng specifically discloses deferring building the index for the corresponding one of the spaced apart bursts of data records after expiration of the corresponding one of the series of spaced apart time intervals as disclosed above and recited at col. 6, lines 48-64 of Cheng's.

Moreover, Cheng further discloses another embodiment [e.g., Cheng: col. 3, lines 4-16; col. 7, lines 18-37; col. 7, lines 59 –col. 8, lines 46] for deferring building indexes after storing bursts of index records in the Small B-Tree (SBT) by periodically using a rolling merge type of procedure (e.g., col. 7, line 21-22) to sparsely populated (e.g., col. 7, lines 29-31) a sufficiently large number of these pointers (or burst of records) of the Small B-Tree (e.g., SBT 150, Fig. 3) in batches (e.g., col. 3, lines 14-16) or in units of pages (col. 8, line 9) into a Large B-Tree (e.g., LBT 132, Fig. 3) of a secondary memory (e.g., the unit 106, Fig. 1).

Therefore, based on the discussion above, in contrary to Appellant's arguments, Examiner concludes that features in claims 9,22 and 37 are fully anticipated by Cheng.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Susan Y Chen
Examiner
Art Unit 2161 7/22/05

Conferees

Charles Rones
SPE – AU 2164

Hosain Alam
SPE – AU 2166

MYERS BIGEL SIBLEY & SAJOVEC
PO BOX 37428
RALEIGH, NC 27627

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Cataloging-in-Publication Data
Microsoft Press Computer Dictionary. -- 3rd ed.
p. cm.
ISBN 1-57231-446-X
1. Computers--Dictionaries. 2. Microcomputers--Dictionaries.
I. Microsoft Press.
QA76.15.M54 1997
004'.03--dc21 97-15489
CIP

Printed and bound in the United States of America.

2 3 4 5 6 7 8 9 QMQM 2 1 0 9 8 7

Distributed to the book trade in Canada by Macmillan of Canada, a division of Canada Publishing Corporation.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office. Or contact Microsoft Press International directly at fax (425) 936-7329.

Acquisitions Editor: Kim Fryer
Project Editor: Maureen Williams Zimmerman, Anne Taussig
Technical Editors: Dail Magee Jr., Gary Nelson, Jean Ross, Jim Fuchs, John Conrow, Kurt Meyer, Robert Lyon, Roslyn Lutsch

bomb[2], crash[2] (definition 1), debug, debugger, hang, inherent error, logic error, semantic error, syntax error. **2.** A recurring physical problem that prevents a system or set of components from working together properly. While the origin of this definition is in some dispute, computer folklore attributes the first use of bug in this sense to a problem in the Harvard Mark I or the Army/University of Pennsylvania ENIAC that was traced to a moth caught between the contacts of a relay in the machine (although a moth is not entomologically a true bug).

buggy \bug´ē\ *adj.* Full of flaws, or bugs, in reference to software. *See also* bug (definition 1).

building-block principle \bil´dēng-blok prin`sə-pəl\ *n. See* modular design.

built-in check \bilt`in chek´\ *n. See* hardware check, power-on self test.

built-in font \bilt`in font´\ *n. See* internal font.

built-in groups \bilt`in grōōps´\ *n.* The default groups provided with Microsoft Windows NT and Windows NT Advanced Server. A group defines a collection of rights and permissions for the user accounts that are its members. Built-in groups are therefore a convenient means of providing access to commonly used resources. *See also* group[1].

bulk eraser \bulk´ ē-rā`sər\ *n.* A device for eliminating all information from a storage medium, such as a floppy disk or a tape, by generating a strong magnetic field that scrambles the alignment of the ferrous materials in the media that encode stored data.

bulk storage \bulk´ stōr`əj\ *n.* Any medium capable of containing large quantities of information, such as tape, fixed disk, or optical disc.

bullet \bōōl´ət, bul´ət\ *n.* A typographical symbol, such as a filled or empty circle, diamond, box, or asterisk, used to set off a small block of text or each item in a list. Round and square bullets are used to set off different levels of information. *See also* dingbat.

bulletin board system \bōōl´ə-tən bōrd si`stəm, bul´ə-tən\ *n. See* BBS.

bulletproof \bōōl´ət-prōōf`, bul´ət-\ *adj.* Capable of overcoming hardware problems that, in another system, could lead to interruption of the task in progress.

bundle \bun´dl\ *vb.* To combine products for sale as a lot. Frequently, operating system software and some widely used applications are bundled with a computer system for sale.

bundled software \bun`dld soft´wâr\ *n.* **1.** Programs sold with a computer as part of a combined hardware/software package. **2.** Smaller programs sold with larger programs to increase the latter's functionality or attractiveness.

burn \burn\ *vb.* To write data electronically into a programmable read-only memory (PROM) chip by using a special programming device known variously as a PROM programmer, PROM blower, or PROM blaster. The term is also used in reference to creating read-only memory compact discs (CD-ROMs). *Also called* blast, blow. *See also* PROM.

burn in \burn in´\ *vb.* **1.** To keep a new system or device running continuously so that any weak elements or components will fail early and can be found and corrected before the system becomes an integral part of the user's work routine. Such a test is often performed at the factory before a device is shipped. **2.** To make a permanent change in the phosphor coating on the inside of a monitor screen by leaving the monitor on and keeping a bright, unchanging image on the screen for extended periods. Such an image will remain visible after the monitor is turned off. Burning in was a danger with older PC monitors; it is no longer a concern with most new PC monitors. *Also called* ghosting.

burst[1] \burst\ *n.* Transfer of a block of data all at one time without a break. Certain microprocessors and certain buses have features that support various types of burst transfers. *See also* burst speed (definition 1).

burst[2] \burst\ *vb.* To break fanfold continuous-feed paper apart at its perforations, resulting in a stack of separate sheets.

burster \bur´stər\ *n.* A device used to burst, or break apart at the perforations, fanfold continuous-feed paper.

burst mode \burst´ mōd\ *n.* A method of data transfer in which information is collected and sent as a unit in one high-speed transmission. In burst mode, an input/output device takes control of a multiplexer channel for the time required to send its data. In effect, the multiplexer, which normally merges input from several sources into a single

high-speed data stream, becomes a channel dedicated to the needs of one device until the entire transmission has been sent. Burst mode is used both in communications and between devices in a computer system. *See also* burst[1].

burst rate \burst´ rāt\ *n.* *See* burst speed (definition 1).

burst speed \burst´ spēd\ *n.* **1.** The fastest speed at which a device can operate without interruption. For example, various communications devices (as on networks) can send data in bursts, and the speed of such equipment is sometimes measured as the burst speed (the speed of data transfer while the burst is being executed). *Also called* burst rate. **2.** The number of characters per second that a printer can print on one line without a carriage return or linefeed. Burst speed measures the actual speed of printing, without consideration of the time taken to advance paper or to move the print head back to the left margin. Almost always, the speed claimed by the manufacturer is the burst speed. By contrast, *throughput* is the number of characters per second when one or more entire pages of text are being printed and is a more practical measurement of printer speed in real-life situations.

bursty \bur´stē\ *adj.* Transmitting data in spurts, or bursts, rather than in a continuous stream.

bus \bus\ *n.* A set of hardware lines (conductors) used for data transfer among the components of a computer system. A bus is essentially a shared highway that connects different parts of the system—including the microprocessor, disk-drive controller, memory, and input/output ports—and enables them to transfer information. The bus consists of specialized groups of lines that carry different types of information. One group of lines carries data; another carries memory addresses (locations) where data items are to be found; yet another carries control signals. Buses are characterized by the number of bits they can transfer at a single time, equivalent to the number of wires within the bus. A computer with a 32-bit address bus and a 16-bit data bus, for example, can transfer 16 bits of data at a time from any of 2^{32} memory locations. Most microcomputers contain one or more expansion slots into which additional boards can be plugged to connect them to the bus.

bus enumerator \bus´ ə-nōō`mər-ā-tər\ *n.* A device driver that identifies devices located on a specific bus and assigns a unique identification code to each device. The bus enumerator is responsible for loading information about the devices onto the hardware tree. *See also* bus, device driver, hardware tree.

bus extender \bus´ eks-ten`dər\ *n.* **1.** A device that expands the capacity of a bus. For example, IBM PC/AT computers used a bus extender to add onto the earlier PC bus and allow the use of 16-bit expansion boards in addition to 8-bit boards. *Also* bus. **2.** A special board used by engineers to raise an add-on board above the computer's case, making it easier to work on the circuit board.

business graphics \biz´nəs graf`iks\ *n.* *See* presentation graphics.

business information system \biz`nəs in-mā´shən si`stəm\ *n.* A combination of computers, printers, communications equipment, and other devices designed to handle data. A computerized business information system receives processes, and stores data; transfers information needed; and produces reports or printouts on demand. *Acronym:* BIS (B`I-S´). *See also* management information system.

business software \biz´nəs soft`wâr\ *n.* A computer application designed primarily for use in business, as opposed to scientific use or entertainment. In addition to the well-known areas of word processing, spreadsheets, databases, and communications, business software for microcomputers also encompasses such applications as accounting, payroll, financial planning, project management, decision and support systems, personnel record maintenance, and office management.

bus mouse \bus´ mous\ *n.* A mouse that connects to the computer's bus through a special circuit board port rather than through a serial port. *Compare* serial mouse.

bus network \bus´ net`wərk\ *n.* A topology (configuration) for a local area network in which all nodes are connected to a main communication line (bus). On a bus network, each node monitors activity on the line. Messages are detected by all nodes but are accepted only by the node to which they are addressed. A malfunctioning node ceases to communicate but does not disrupt